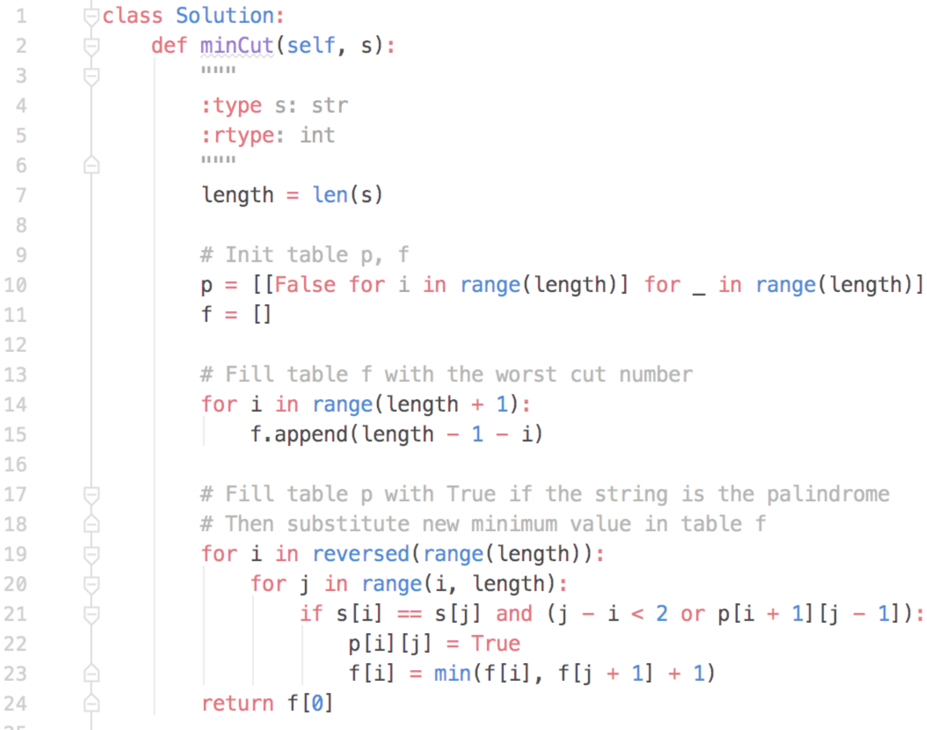
P: Initial state			
i\j	0	1	2
0	F	F	F
1	F	F	F
2	F	F	F

F: Initial state, worst cut				
i	0	1	2	3
F(i)	2	1	0	-1



```python
class Solution:
    def minCut(self, s):
        """
        :type s: str
        :rtype: int
        """
        length = len(s)

        # Init table p, f
        p = [[False for i in range(length)] for _ in range(length)]
        f = []

        # Fill table f with the worst cut number
        for i in range(length + 1):
            f.append(length - 1 - i)

        # Fill table p with True if the string is the palindrome
        # Then substitute new minimum value in table f
        for i in reversed(range(length)):
            for j in range(i, length):
                if s[i] == s[j] and (j - i < 2 or p[i + 1][j - 1]):
                    p[i][j] = True
                    f[i] = min(f[i], f[j + 1] + 1)
        return f[0]
```

i	2	1		0		
j	2	1	2	0	1	2

P: Final state			
i\j	0	1	2
0	T	T	F
1	F	T	F
2	F	F	T

f[i] = min(f[i] , f[j + 1])				
i\j	0	1	2	3
0	2	1		
1		1		
2			0	
3				